

PARTIAL DIFFERENTIAL EQUATION MODEL FOR IMAGE
FEATURE EXTRACTION AND IDENTIFICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

This patent application claims the benefit of the filing date of United States Provisional Patent Application Serial No. 60/317,219, filed September 5, 2001 and entitled "ENHANCED IMAGE MODELING"; the entire contents of which are hereby expressly incorporated by reference.

FIELD OF THE INVENTION

The present invention relates to image processing. More specifically, the invention relates to numerical modeling of biometric features, such as fingerprints.

BACKGROUND OF THE INVENTION

Fingerprint identification is one of the most important biometric related technologies. A fingerprint of a person comprises a distinctive and unique ridge pattern structure. For authentication or identification purposes, this ridge pattern structure can be characterized by endings and bifurcations of the individual ridges which is popularly known as minutiae. As a result, the accuracy of minutia extraction is crucial in the overall success of fingerprint authentication or identification.

Typically, in a good quality fingerprint image, 70-100 minutiae can be located precisely. But, in a poor quality fingerprint image, the number of minutiae that are surely and steadily extractable by common feature extraction algorithms is much less (approximately 20-50). A well-designed enhancement algorithm can dramatically improve the extraction of minutiae.

Usually, ridge pattern detection is performed manually by professional fingerprint experts. However, manual detection is tedious, time-consuming, and expensive and does not meet the performance requirements of the newly developed applications.

1 Most of the automatic fingerprint feature extraction methods
employ conventional image processing techniques for fingerprint
feature extraction and suffer from noisy artifacts of the input
5 fingerprint image in practice. Some research in fingerprint
image enhancement have been reported, for example, in L. Hong,
A. K. Jain, S. Pankanti, and R. Bolle, "Fingerprint Enhancement",
Proc. First IEEE WACV, pp. 202-207, Sarasota, Fla., 1996; P. E.
Danielsson and Q. Z. Ye, "Rotation-Invariant Operators Applied
10 to Enhancement of Fingerprints", Proc. Ninth ICPR, pp 329-333,
Rome, 1988; and D. C. Huang, "Enhancement and Feature
Purification of Fingerprint Images", Pattern Recognition, Vol.
26, no. 11, pp. 1221-1671, 1993; the contents of which are hereby
incorporated by reference.

15 However, most of the published approaches for fingerprint
image enhancement use conventional image processing technology
to improve the clarity of ridge structures. Common fingerprint
feature extraction algorithms employ image-processing techniques
to detect minutiae. These techniques adopt only a bottom-up
20 computational paradigm, in which no high level knowledge about
fingerprint property is used to guide the processing.

Therefore, there is a need for an accurate and efficient
technique for generating a geometric pattern based on visual
appearances of a biometric image.

25 SUMMARY OF THE INVENTION

The present invention discloses a new approach for
automatically generating a geometric pattern based on local
visual appearances of an image, such as a fingerprint, facial
30 images, eye iris or retina images, or any other flow-like images
such as, texture images and the like. The basic idea is
considering ridge patterns in the fingerprint as flow-like
patterns. A second order differential equation representing the
flow of fluid, is employed in describing and enhancing the
35 different local shapes of the fingerprint. A special kind of

1 partial differential equation is formed by transferring the
original values of the fingerprint image to corresponding
coefficients of the partial differential equation. The partial
5 differential equation is solved according to a boundary condition
that is determined by a local region. Therefore, a relationship
between feature extraction of fingerprint and the partial
differential equation is established. An enhanced image that
reveals the ridge patterns of the fingerprint is obtained by
10 mapping back the solutions of the partial differential equation
into the corresponding positions in the image.

Since fingerprints are graphical flow-like ridges present
on human fingers, it is possible to view the local patterns of
a fingerprint as surface appearances of fluid flow and employ a
15 mathematical model to accomplish enhancement and feature
extraction. The present invention establishes a mathematical
model according to local regions conditions in the fingerprint
image, converts the model into numerical equations for processing
by a computer, solves the numerical equations, and transfers the
20 solutions back into the local regions of the image. Experimental
results show that a model-based enhancement improves the clarity
of ridge structures of fingerprint image. As a result,
fingerprint feature extraction can be achieved at a more accurate
and robust level.

25 In one aspect, the invention is a method performed by a
computer for extracting features from an image, the method
comprising the steps of: establishing a mathematical model
according to regional conditions in the image; converting the
mathematical model into numerical equations; solving the
30 numerical equations; and transferring the solutions of the
numerical equations to respective regions of the image.

In another aspect, the invention is a digital signal
processor (DSP) storing a set of instructions for generating
geometric pattern from an image having a plurality of ridges and
35 mesh points. When executed, the instructions cause the DSP to

1 perform the steps of: forming a partial differential equation by
transferring values for positions in the image to corresponding
coefficients of the partial differential equation; determining
5 simultaneous difference equations corresponding to the partial
differential equation and the image mesh points; solving the
simultaneous difference equations; and mapping the solutions of
the simultaneous difference equations to respective positions on
the image to determine features of the image.

10 BRIEF DESCRIPTION OF THE DRAWINGS

The objects, advantages and features of this invention will
become more apparent from a consideration of the following
detailed description and the drawings, in which:

15 FIG. 1 is an exemplary overall flow diagram showing the
common stages in a fingerprint identification system;

FIG. 2 is an exemplary diagram showing exemplary steps in
fingerprint feature extraction;

FIG. 3 is an exemplary good-quality fingerprint image;

20 FIGS. 4A and 4B are exemplary views showing two commonly
used fingerprint features: A. ridge bifurcation and B. ridge
ending;

FIGS. 5A and 5B are exemplary views showing complex features
of a fingerprint image as a combination of simple features: A.
25 short ridge and B. enclosure;

FIG. 6 shows an example of a local region taken from a
fingerprint image.

30 FIG. 7 is an exemplary flow chart for fingerprint feature
extraction for a local region, according to one embodiment of the
present invention;

FIG. 8 is an exemplary diagram for setting partial
differential equation, according to one embodiment of the present
invention;

1 FIG. 9 is an exemplary diagram for getting simultaneous
difference equations, according to one embodiment of the present
invention;

5 FIG. 10A is an exemplary drawing of two adjacent discrete
points for discretization of first derivatives along with X-
direction, according to one embodiment of the present invention;

10 FIG. 10B is an exemplary drawing of two adjacent discrete
points for discretization of first derivatives along with Y-
direction, according to one embodiment of the present invention;

15 FIG. 10C is an exemplary drawing of three adjacent discrete
points for discretization of second derivatives along with X-
direction, according to one embodiment of the present invention;

20 FIG. 10D is an exemplary drawing of three adjacent discrete
points for discretization of second derivatives along with Y-
direction, according to one embodiment of the present invention;

25 FIG. 10E is an exemplary drawing of four adjacent discrete
points for discretization of mixed derivatives, according to one
embodiment of the present invention;

30 FIG. 11A is an exemplary drawing for performing first and
second derivatives discrete by selecting referent points as
adjacent points near a boundary line, according to one embodiment
of the present invention;

35 FIG. 11B is an exemplary drawing for mixing derivatives
discrete by selecting referent points as adjacent points near a
boundary line, according to one embodiment of the present
invention;

40 FIG. 12A illustrates the relationship between a continuous
boundary and a discrete boundary;

45 FIG. 12B illustrates the discretization of the continuous
and differentiable function on a numerical boundary;

50 FIG. 13 is an exemplary flow chart showing the steps
performed for solving a SDE, according to one embodiment of the
present invention;

1 FIG. 14 shows an interpolated point and its 3*3 neighboring mesh points;

5 FIG. 15 illustrates the operation of mapping a SDE solution back into an image, according to one embodiment of the present invention;

 FIG. 16 shows an exemplary result of local solution mapping; and

10 FIG. 17 shows an exemplary enhanced image obtained by mapping a solution to the entire area of an original fingerprint image.

DETAILED DESCRIPTION

15 The present invention establishes numerical relationship between visual appearance of a biometric image and approximate solution of a partial differential equation with a boundary condition. This new approach differs from current methods for the fingerprint image processing. Conventional approaches employ image processing techniques for enhancement and feature extraction. In contrast, the present invention builds and
20 utilizes a mathematical model of a local shape of the biometric image, establishes the relationship between local region status and partial differential equation, and then determines the features of the local shape by solving the numerical equations.

25 A fingerprint identification system includes various processing stages as shown in FIG. 1. For an automatic fingerprint identification system, a precise and robust feature extraction is essential. Feature extraction can be divided into two main operations: (a) image enhancement and (b) feature
30 selection, as shown in FIG. 2. The critical operation in feature extraction is the image enhancement due to the presence of noise in the fingerprint image. Although a fingerprint is used as an example for simplicity purposes, the techniques of the present invention can be used for other biometric images, such as palm
35

1 images, facial images, eyes iris images, or any other flow-like
images such as texture images and the like.

5 The performance of most commercially available feature
extraction algorithm relies heavily on the quality of the input
fingerprint images. In an ideal fingerprint image, the ridge
structure can be easily detected and minutia features can be
precisely located from the image. FIG. 3 shows an example of good
quality fingerprint image.

10 However, in practice, due to variations in acquisition
devices, skin conditions, etc., most of the acquired fingerprint
images are of poor quality. For example, the ridge structures
are not always well defined and therefore they cannot be
correctly extracted. In spite of rich information included in
15 a fingerprint, only two kinds of features are preferably selected
for identification or authentication in most current available
systems. In one embodiment, the set of fingerprint features is
restricted to two types of minutiae: bifurcations and ridge
endings. Some examples of bifurcations and ridge endings are
20 shown in FIGs. 4(A) and 4(B), respectively. More complex
fingerprint features can be expressed as a combination of these
two basic features. FIG. 5(A) gives an example of a short ridge
that can be considered as a pair of ridge endings and FIG. 5(B)
shows an enclosure that can be considered as a collection of two
25 bifurcations.

Since a minutia is a local feature of a fingerprint, the
fingerprint image may be segmented into local regions. The size
of each local region is selected to allow for at least two ridges
in the region. Theoretically, there is no limitation on the
30 shape of local regions. As shown in FIG.6, for simplicity, a
square region of 64*64 pixels size is cut out from the original
fingerprint image to describe the following steps of the present
invention.

35 The method of the present invention, as described below, is
independent of the position of the local region. That is, the

1 same steps can be performed on each local region of the
fingerprint image for processing the entire fingerprint image.
However, for simplicity reasons, the following description is
related to one (local) region of the fingerprint image.

5 FIG. 7 illustrates an exemplary flow chart for enhancing and
extracting features from each local region. In block 701, some
intrinsic properties of the fingerprint image are calculated
according to the local ridge pattern. Then, in block 702, a
10 partial differential equation is established by mapping the
intrinsic properties into the coefficients of the equation and
a boundary condition is determined. Subsequently,
integralization, discretization and transformation steps are
performed on the image. Each derivative and variable items of
15 the partial differential equation with respect to each mesh
points in local region of the fingerprint is replaced so as to
obtain corresponding simultaneous difference equations, as shown
in block 704.

20 A numerical method is then employed to solve the
corresponding simultaneous difference equations in block 706.
In block 708, the solution is mapped back into the local region
of the fingerprint image. Finally, the fingerprint shape and
feature are determined by using normalization and interpolation
processes, as depicted in block 709.

25 Based on theoretical study of local patterns in fingerprint
images, a mathematical model is established in terms of partial
differential equation (PDE) as follow:

$$A1 \frac{\partial^2 U}{\partial X^2} + A2 \frac{\partial^2 U}{\partial X \partial Y} + A3 \frac{\partial^2 U}{\partial Y^2} + A4 \frac{\partial U}{\partial X} + A5 \frac{\partial U}{\partial Y} + A6 * U = 0 \quad (1)$$

30 Where A1, A2, A3, A4, A5 and A6 are weight coefficients of
the partial differential equation.

Theoretical research and experimental results demonstrate
that such model sufficiently represents almost any local shapes
35 that appear in fingerprint images. In order to solve such

1 special kind of partial differential equation, an initial
condition and a boundary condition should be given based on a
local region.

5 FIG. 8 is an exemplary flow chart showing steps of setting
initial condition and boundary condition. Five values related
to intrinsic properties of the local region are calculated to
initialize the PDE (block 804). Then, the weight coefficients
of the partial differential equation are determined by the
10 intrinsic property values.

The five values M, V, P, Q and W are defined as follows:

M and V denote the estimated mean and variance of the gray-
level values in the local region respectively.

P and Q are the first and second components of the local
ridge oriental vector respectively.

15 W is the local ridge frequency.

Since local gray-level values vary in different region of
fingerprint image, the local region should be normalized (block
808) so that correct mean and variance of local region can be
obtained. The steps involved in normalizing algorithm are as
20 follows:

1. The mean of the gray-level in local region R, is
calculated as

$$25 \quad M = (1/N) \sum_{(I,J) \in R} F(I,J) \quad (2a)$$

Where N is total number pixels in region R; F(I, J) is gray
value of the fingerprint image at point (I, J).

30 2. The variance of region R is defined as

$$V = (1/N) \sum_{(I,J) \in R} (F(I,J) - M)^2 \quad (2b)$$

1 3. A normalized region R is determined using the following equation:

$$\begin{aligned} R(I,J) &= m + \sqrt{(v * (F(I,J) - M) * (F(I,J) - M)) / V), \text{ if } (I,J) > M; \\ R(I,J) &= m - \sqrt{(v * (F(I,J) - M) * (F(I,J) - M)) / V), \text{ otherwise} \end{aligned} \quad (2c)$$

5 where m and v are the desired mean and variance values, respectively.

As described above, normalization is a pixel-wise operation. The purpose of normalization is to reduce the variations in gray-level values along ridges and valleys, which facilitates the subsequent processing. Due to the presence of noise, smudges, and breaks in ridges in the input fingerprint image, the estimated local property values, P, Q, and V, may not be accurate or stable. Typical fingerprint smooth operator such as histogram modeling described in Anil K. Jain, Fundamentals of Digital image processing, Prentice Hall, Englewood Cliffs, NJ, 1989; and median filter described in Jae S. Lim, Two-dimensional Signal and Image Processing, Prentice Hall, Englewood Cliffs, NJ, 1990, the entire contents of which are hereby incorporated by reference, can be used to modify the local region.

By viewing a fingerprint image as an oriented texture, a number of methods have been proposed to estimate the ridge oriental vector of fingerprint images. For example, see, M. Kass and A. Witkin, "Analyzing oriented Patterns", Computer Vision, Graphics, and Image Processing, vol. 37, no. 4, pp. 362-385, 1987; and A. Rao, A Taxonomy for Texture Description and Identification. New York, NY: Springer Verlag, 1990; the entire contents of which are hereby incorporated by reference. The main steps of ridge oriental vector estimating algorithm for estimating the property values p and Q at local region of fingerprint image, shown in block 810, are as follows:

1. Divide local region R into blocks of size b*b.
2. Compute the gradients at each pixel in R. Let $\partial_x(I, J)$ and $\partial_y(I, J)$ be the gradient magnitude in x and y

directions, respectively, at pixel (I, J) of the image. The gradients are defined as

$$\begin{aligned}\partial x(I,J) &= (p_1 * F(I-d,J) + p_2 * F(I,J) + p_3 * (F(I+d,J))) / p, \\ \partial y(I,J) &= (p_1 * F(I,J-d) + p_2 * F(I,J) + p_3 * (F(I,J+d))) / p\end{aligned}\quad (3a)$$

Where p_1 , p_2 are positive numbers and p_3 is negative number, d is a constant expressed as step of the gradients.

$$p = p_1 + p_2 + p_3;$$

3. Estimate the local orientation of each block $B(k)$ centered at pixel (I, J) using the following equation:

$$\xi_x(I,J) = \sum_{(u,v) \in B(k)} \sum 2 * \partial x(u,v) * \partial y(u,v), \quad (3b)$$

$$\xi_y(I,J) = \sum_{(u,v) \in B(k)} \sum (\partial^2 \partial x(u,v) * \partial^2 \partial y(u,v)), \quad (3c)$$

$$\theta(I,J) = (1/2) \text{atan}\{ \xi_x(I,J) / \xi_y(I,J) \} \quad (3d)$$

Where $\theta(I,J)$ is an estimate of the local ridge orientation at the block centered at pixel (I, J).

4. Compute the ridge oriental vector using

$$P = (1/n) \sum_{(I,J) \in R} \cos(2 * \theta(I,J)) \quad (4a)$$

$$Q = (1/n) \sum_{(I,J) \in R} \sin(2 * \theta(I,J)) \quad (4b)$$

Where n is the total number of pixels calculated at the local region R .

1 With above algorithm, a fairly smooth orientation estimate
can be obtained. The weight numbers p_1 , p_2 , p_3 and step d are
pre-determined numbers based on the block size b and the
5 resolution of fingerprint image. Typically, fingerprint images
are scanned at resolution of 500 dots per inch (dpi). In this
case, the block size is defined as 5×5 , and $p_1 = p_3 = 1$, $p_2 = -2$
and $d = 1$.

Several techniques can be used for oriental vector
10 computing. Depending on the computational requirement, the
gradient operator may vary from the simple Sobel operator
described in Sanjit K. Mitra and James F. Kaiser "Handbook for
Digital Signal Processing", A Wiley-Interscience Publication.
John Wiley & Sons, 1993 to a more complex Marr-Hildreth operator
15 described in D. Marr, Vision: A Computational Investigation into
the Human Representation and Processing of Visual Information.
W.H. Freeman and Company. New York, 1982, the contents of which
are hereby incorporated by reference.

Local ridge frequency is defined as frequency of ridge and
20 valley structures in a local neighborhood along the direction
normal to the local ridge orientation. Local ridge frequency is
another intrinsic property of a fingerprint image. The steps
involved in local ridge frequency estimation W are as follows:

1. Divide the local region R into small windows of size
25 $b \times b$. Denote the window centered at pixel (I, J) as
 $wnd(I, J)$.
2. For each window centered at pixel (I, J) , estimate the
main oriental vector (p, q) .
3. For each window centered at pixel (I, J) , compute the
30 minimal value and maximal value within a block of
appropriate size, which depend directly on the
resolution of the fingerprint image. In practice, as
fingerprint images are scanned at resolution of 500
dpi, the size of block can be fixed at 10×10 .

4. For each window centered at pixel (I, J), get a sequence of pixels that take minimal and maximal value along the direction (a, b). Where (a, b) is orthogonal vector of the main oriental vector (p, q). Denote the sequence of pixel as seq(I, J).
5. Calculate the frequency of seq(I, J) at each window centered at pixel (I, J) according to the differential value between connected elements in seq(I, J). Denote the frequency of seq(I, J) as freq(I, J).
6. Estimate the local ridge frequency by following equation:

$$W = (1/K) \sum_{(u,v) \in \text{wnd}(I,J)} \text{freq}(u,v) \quad (5)$$

Frequency estimation is a window-wise operation. As described above, local ridge frequency W is obtained as the average frequency of the sequence frequency in a window. When the window to be processed contains corrupted ridges and valleys, the sequence of pixels that appear maximal or minimal value need to be well selected. In some cases, interpolations need to be performed on pixel sequence obtained in that window.

Once the values M, V, P, Q and W corresponding to the fingerprint intrinsic properties of the local region R are determined, a transformation that maps local intrinsic properties M, V, P, Q and W into weight coefficients of a special partial differential equation is performed, as illustrated in block 812. As a result, a partial differential equation (1) that describes the ridge pattern of the partial fingerprint image is formed. The weight coefficients, A1, A2, A3, A4, A5 and A6 are determined by the following transformation:

$$A1 = P * P * (P * P + Q * Q) * W * W, \quad (6a)$$

$$A2 = 2 * (\text{sqrt}(u * u - P * P * W * W) * \text{sqrt}(v - Q * Q * W * W)) / W, \quad (6b)$$

$$A3=Q*Q(P*P+Q*Q)*W*W, \quad (6c)$$

$$A4=u*q+v, \quad (6d)$$

$$A5=-v*p-u, \quad (6e)$$

$$A6=a*(P*P+Q*Q)+b \quad (6f)$$

Where a, b, u, v are constants, specified as initial parameters at system setup.

Well-selected constants improve the performance of overall system. In one implementation, a simulation of the partial differential equation for setting optimal initial parameters for system and evaluation of results shows that the selection of constants involves a trade-off between complexity and accuracy. That is, the more optimal the parameters, the more sophisticated ridge information may be created but, the partial differential equation would be more prone to noise. On the other hand, the less optimal the parameters, the less noise prone the system is and less computation time is required. However in this case, detailed ridge shape loss increases.

The PDE is completely determined by intrinsic properties of the local region as its boundary conditions. In order to create boundary conditions for the PDE, a close boundary is drawn out manually, or by a computer routine within the respective local region, as shown in block 814. The boundary conditions are then set in block 816. For example, B is denoted as a boundary in local region R. The partial differential equation (1) including the following boundary condition expresses a solvable mathematical problem.

$$U \Big|_{(x,y) \in S} = B1(x,y), \quad \frac{\partial U}{\partial s} \Big|_{(x,y) \in S} = B2(x,y) \quad (7)$$

Where S is a continuous boundary defined on the discrete boundary B, $B1(x, y)$ and $B2(x, y)$ are the continuous function and differentiable function defined on the boundary S, respectively.

As mentioned above, the enhancement of local ridge shape of fingerprint is considered as the solution of the corresponding PDE under boundary condition. So far, the relationship between a special partial differential equation which governs a fluid flow phenomenon and local intrinsic properties of a fingerprint image is established and the boundary condition related to the gray-level variations of local region in the fingerprint is determined. Next, the partial differential equation needs to be solved with the given boundary condition and the solution to be applied to the local regions of the fingerprint.

In order to numerically solve the PDE in equation (1) with given boundary condition (7) by a computer, three processes namely, integralization, discretization and transformation are performed on the respective local region with the PDE and the boundary condition. FIG. 9 is an exemplary block diagram showing overall procedures involved in solving the PDE (block 706 of FIG. 7).

Integralization step of block 904 produces a group of integral points within the local region R and an integral boundary IB (block 912):

1. Let's denote two directions of the coordinate axes of the fingerprint image as X-direction and Y-direction. Along with X-direction and Y-direction, integral points carry out at a desired step length H as follow:

$$X(I) = X0 + I*H, \quad I = 0, 1, 2, \dots, W(F); \quad (8a)$$

$$Y(J) = Y0 + J*H, \quad J = 0, 1, 2, \dots, H(F). \quad (8b)$$

Where, $(X0, Y0)$ is top left point of the image, $W(F)$ is the width of the image and $H(F)$ is the height of the image.

2. Map the above integral points into the local region R. Refer to the integral points within region R as inner mesh points of R. Denote the overall inner mesh points of R as IMP(R).

3. An integral boundary IB is formed by selecting the nearest inner mesh points of R.

When the integralization process for inner mesh points and mesh points near boundary line is completed, a discretization based on mesh points is performed for numerating the partial differential equation (1) and the boundary condition (7), as shown in block 906. First, derivatives are replaced with respect to each inner mesh point (block 914), within the original partial differential equation (1), for obtaining a corresponding numerical equation. The replacing rules are defined as follow:

$$\frac{\partial U}{\partial X} \approx [U(X+H, Y) - U(X, Y)]/H \quad (9a)$$

$$\frac{\partial U}{\partial Y} \approx [U(X, Y+H) - U(X, Y)]/H \quad (9b)$$

$$\frac{\partial^2 U}{\partial X^2} \approx [U(X+H, Y) - 2*U(X, Y) + U(X-H, Y)]/(H*H) \quad (9c)$$

$$\frac{\partial^2 U}{\partial Y^2} \approx [U(X, Y+H) - 2*U(X, Y) + U(X, Y-H)]/(H*H) \quad (9d)$$

$$\frac{\partial^2 U}{\partial X \partial Y} \approx [U(X+H, Y+H) - U(X+H, Y) - U(X, Y+H) + U(X, Y)]/(H*H) \quad (9e)$$

Where (X, Y) is inner mesh point in IMP(R).

Note that each derivative included in the original partial differential equation (1) is converted into a discretized

equation. In other words, the numerical equation is the result of discretization based upon the integration of inner points within the boundary B. The first derivative of the partial differential equation is expressed as a difference of two adjacent discrete points, as shown in FIGs. 10(a) and 10(b). The second derivative of the equation includes three adjacent discrete points, as shown in FIGs. 10(c), and 10(d). The mix derivative (9e) depends on four discrete points, as shown in FIG. 10 (e). In some cases, especially in the location near the boundary, an inner mesh point may not have enough adjacent points to support derivatives replacement. Thus, neighboring points outside of local region should be selected as adjacent points. FIG. 11(a) gives an example of first and second derivatives select neighboring points as adjacent points. FIG. 11(b) shows a mix derivatives replacement supporting by three neighboring points.

Second, similar replacing rules can be used to convert the boundary condition (7) into a numerical condition with respect to each integral point of boundary IB, as shown in block 916. The relationship between a continuous boundary S and a numerical boundary IB is shown in FIG. 12(a). In the original boundary condition (7), the continuous function $B1(x, y)$ defined on the continuous boundary S is replaced with a numerical function $D1(X, Y)$ defined on a numerical boundary IB. The numerical function $D1(X, Y)$ is carried out by following expression:

$$D1(X,Y)=f1*F(X,Y)+f2, \quad (X,Y) \in IB \quad (10)$$

Where $f1$ and $f2$ are constants that are predetermined according to the brightness and contrast of the image and $F(X, Y)$ is the gray value at point (X, Y) on the integral boundary IB. The differentiable function $B2(x, y)$ defined on the continuous boundary S is replaced with a numerical function $D1(X, Y)$ defined

on a numerical boundary IB. The numerical function $D2(X, Y)$ is carried out by following expression:

$$\begin{aligned} D2(X,Y) &= f1 * [F(X1,Y1) - F(X,Y)] / h, & (X,Y) \in IB \\ h &= \text{sqrt}((X1-X)*(X1-X) + (Y1-Y)*(Y1-Y)); \end{aligned} \quad (11)$$

Where $(X1, Y1)$ is an integral point on IB which is selected as the next adjacent point along the boundary line IB. FIG. 12(b) is an exemplary diagram for explaining the discretization of the continuous and differentiable function on a numerical boundary IB. Finally, combining the numerical derivatives of the partial differential equation and numerical boundary condition (block 918), the original partial differential equation (1) with the boundary condition (7) is expressed as an approximate numerical partial differential equation with numerical boundary condition using the following replacements:

1. Each first derivative included in the original partial differential equation (1) is replaced with the discrete form according to the expression (9a) and (9b).
2. Each second derivative included in the original partial differential equation (1) is replaced with the discrete form according to the expression (9c) and (9d).
3. The mix derivative in the original partial differential equation (1) is replaced with the discrete form according to the expression (9e).
4. The continuous function $U(X, Y)$ in the boundary condition (7) is replaced with the discrete function $F(I, J)$ which (I, J) is inner mesh point of local region.
5. The continuous function $B1(x, y)$ and the differentiable function $B2(x, y)$ in the boundary condition(7) is replaced with a numerical functions

1 according to the expression (10) and (11),
 respectively.
 6. Combing the same items and reducing the void
 5 expression, the original PDE (1) with (7) is expressed
 as a discrete formula:

$$\begin{aligned} C1*U(I+H,J+H)+C2*U(I+H,J)+C3*U(I,J+H)+ \\ C4*U(I-H,J)+C5*U(I,J-H)+C6*U(I,J)=0; \end{aligned} \quad (12)$$

$$\begin{aligned} 10 \quad U(I0,J0)=f1*F(I0,J0)+f2 \text{ and} \\ U(I1,J1)-U(I0,J0)=(F(I1,J1)-F(I0,J0))/h. \end{aligned} \quad (13)$$

15 Where (I, J) is a inner mesh point of the local region of
 the fingerprint image, (I0, J0) and (I1, J1) are adjacent points
 along the numerical boundary line, H is a predetermined step
 length and h is the distance between two points (I0, J0) and (I1,
 J1). The coefficients C1, C2, C3, C4, C5 and C6 are determined
 by following expressions:

$$20 \quad C1=A2/(H*H); \quad (14a)$$

$$C2=[A1/H*H)-A2/(H*H)+A4/H]; \quad (14b)$$

$$C3=[A3/(H*H)-A2/(H*H)+A5/H]; \quad (14c)$$

$$C4=A2/(H*H); \quad (14d)$$

$$25 \quad C5=A3/(H*H); \quad (14e)$$

$$C6=[A2/(H*H)-2*A1/(H*H)-2*A3/(H*H)-A4/H-A5/H+A6]. \quad (14f)$$

30 Eventually, the original PDE (1) with boundary condition (7)
 is completely converted to a discrete representation (12) under
 discrete boundary condition (13). At each mesh point (I, J) in
 local region, executing the discretization produce at least one
 equation as expressed as (12) about six neighboring points (I+H,
 J+H), (I+H, J), (I, J+H), (I-H, J), (I, J-H), (I, J). On the
 discrete boundary, more equations would be produced.

35

1 Referring to block 908, through following transformation,
 a simultaneous discrete equation, commonly referred to as
 simultaneous difference equations (SDE), is formed. Considering
 5 each mesh point value $U(I, J)$ in local region as variable to be
 solved, a group of equations about mesh point values are obtained
 as expressed as (12) and (13). Putting such equations together
 and departing the coefficients from the variables, the equations
 are transferred into simultaneous format that is expressed as
 10 follows.

$$A\{U\} = c \quad (15)$$

15 Where A is coefficients item, U is variables item and c is
 a constant item.

According to the numerical operations described above, the
 simultaneous difference equations to be handled are limited to
 the orthogonal mesh points which spread over a given region, and
 the numerical operations (referred to as integralization,
 20 discretization and transformation) are calculated only for the
 mesh points. That is, the simultaneous difference equations can
 be modified only in the position where the image points are
 selected as inner mesh points. For such reason, preferably an
 optimal mesh point selection should be done before the numerical
 operation. The step length H is flexible and can be set
 25 according to different criteria. For example, longer step length
 may be chosen at a region where the frequency of the ridge
 pattern is low, or shorter step length may be chosen at higher
 ridge frequency region to generate more inner mash point.

30 The more inner mesh points are generated, the more discrete
 equations are transformed into the simultaneous difference
 equations. As a result, the simultaneous difference equations
 to be analyzed become larger in size and the computation time
 required for solving these equations increases in proportion to
 35 the size of the equations. Taking the computation time and

1 memory requirement into account, a variable step length can be
 chosen so that the simultaneous difference equations can be
 optimally determined based on different scenarios.

5 By performing the integralization, discretization and
 transformation steps in block 920, the original partial
 differential equation (1) with boundary condition (7) is replaced
 by the simultaneous difference equations (SDE) that includes
 discrete values. That is, the original equation (1) with (7) is
 10 approximated as a group of discrete equations. The numerical
 solution of the SDE may be carried out by a general-purpose
 computer. FIG. 15 depicts an exemplary flow chart of the steps
 performed for solving the SDE.

15 Referring back to block 706 of FIG.7, several conventional
 methods such as finite element method can be employed to solve
 the simultaneous difference equations. Considering the style and
 limitation of the simultaneous difference equations, Galerkin's
 method (described in J.N.Reddy. "An Introduction to the Finite
 Element Method" McGraw-Hill, Inc., 1993, the contents of which
 20 are hereby incorporated by reference) is employed to solve the
 simultaneous difference equations. The detailed description of
 Galerkin's method is beyond the scope of this invention. As a
 reference, main steps of Galerkin's method are described as
 following:

- 25 1. Express the simultaneous difference equations in a
 standard form as
- $$A\{U\} = f \quad (16)$$

Where A is a coefficient item and U is the object item.

- 30 2. Select a complete element sequence $\{V(k)\}$ ($k = 1, 2,$
 $3, \dots, n$), and an appropriate number n.
 3. Take an approximate solution which expresses as

35
$$U(n) = \sum a(k) * V(k) \quad (17)$$

1 Where $a(k)$ are constants to be determined.

Insert the approximate solution into the expression
(16) and multiple both sides of the expression by $V(k)$
5 so that simultaneous algebra equations with respect to
 $a(k)$ are obtained as

$$\sum a(k) \cdot (A \cdot V(k), V(j)) = (f, V(j)) \quad (j=1, 2, \dots, n) \quad (18)$$

10 4. Get the solution of $a(k)$ and replace back into the
expression (16), the approximate solution $U(n)$ is obtained.

As described above, when a point in the fingerprint image
is selected as a mesh point and discretization is achieved, an
enhanced value can be approximated by using Galerkin's method.
15 In a case where the image position is not on the mesh point,
interpolation is used to calculate the value. In other words,
the enhanced value at a position other than a mesh point is
interpolated based on the value of the neighboring mesh points.
An exemplary process of interpolation is described as follow:

20 1. For each point (x, y) which is not selected as a mesh
point in the image, get the neighboring mesh points
sequence:

$$P(i, j) = (x(i), y(j)) \quad i=1, 2, \dots, m; \quad j=1, 2, \dots, n.$$

25 The corresponding value sequence is expressed as:

$$\{v(i, j)\} \quad i=1, 2, \dots, m; \quad j=1, 2, \dots, n.$$

2. Set a Lagrange polynomial with respect to x as:

$$\alpha(x, i) = \omega(x) / ((x - x(i)) \cdot \tau(x(i))) \quad i=1, 2, \dots, m \quad (19a)$$

30 Where $\omega(x)$ and $\tau(x(i))$ are defined as

$$\omega(x) = \prod (x - x(i)) \quad (19b)$$

$$\tau(x(i)) = (x(i) - x(1)) \cdot \dots \cdot (x(i) - x(i-1)) \cdot (x(i) - x(i+1)) \cdot \dots \cdot (x(i) - x(n)) \quad (19c)$$

35

3. Set a Lagrange polynomial with respect to y as:

$$\beta(y,j)=\omega(y)/((y-y(j))*\tau(y(j)) \quad j=1,2,\dots,n \quad (19d)$$

Where $\omega(y)$ is defined as

$$\omega(y)=\prod (y-y(j)) \quad (19e)$$

and $\tau(y(j))$ is defined as

$$\tau(y(j))=(y(j)-y(1))*\dots(y(j)-y(j-1))*(y(j)-y(j+1))*\dots(y(j)-y(n)) \quad (19f)$$

4. Calculate the value at point (x, y) as

$$v(x,y)=\sum \sum v(i,j)*\alpha(x,i)*\beta(y,j) \quad (19g)$$

Accordingly, there may be a variation in the interpolation precision depending on the manner the neighboring mesh points sequence is obtained. In one embodiment, neighboring mesh points sequence is quadratically selected and both size m and n are set to 3. FIG. 14 shows an interpolated point and its 3*3 neighboring mesh points.

After applying the above steps to all the local regions of the fingerprint image, a solution for each pixel in the image is obtained with different precision. The gray level precision of image for computer processing is typically set to 8-bit (namely a byte) in which the minimum gray value is 0 and maximum gray value is 255. A normalization process described as follow is then used to map the solution back into the image.

FIG. 15 illustrates the operation of mapping the SDEs solution back into the image (block 1502). As depicted in block 1508, for each local region R, minimum value and maximum value is calculated in the solution, and the minimum and maximum value denoted as min (R) and max (R), respectively. In block 1510, the ratio value $r(R)=255/(max(R)-min(R))$ is obtained. For each

1 point (I, J) in the region R, the solution at point (I, J) is
denoted as $v(I, J) * 3$. In block 1512, $w(I, J)$ is mapped into
gray level byte at the position (I, J) as $W(I, J) = r(R) * (v(I,$
5 $J) - \min(R))$.

The local region R is then enhanced by placing the value
 $W(I, J)$ at position (I, J). The size of local region R in
normalization is set in such a manner to cover at two ridges.
FIG. 16 shows an exemplary result of local solution mapping.
10 FIG. 17 shows an exemplary enhanced image that is obtained by
mapping the solution to the entire original fingerprint image.

The present invention may be implemented by a computer
software programmed to perform the steps involved. Alternatively
a Digital Signal Processor (DSP) chip may be programed to perform
the steps of the present invention. In another embodiment, a
15 special electronic circuit, such as an Integrated Chip (IC),
maybe designed to carry out the steps of the present invention.

It will be recognized by those skilled in the art that
various modifications may be made to the illustrated and other
embodiments of the invention described above, without departing
from the broad inventive scope thereof. It will be understood
20 therefore that the invention is not limited to the particular
embodiments or arrangements disclosed, but is rather intended to
cover any changes, adaptations or modifications which are within
the scope of the invention, as defined by the appended claims.
25